

DIALOG(R)File 351:Derwent WPI  
(c) 2003 Thomson Derwent. All rts. reserv.

011674753      \*\*Image available\*\*

WPI Acc No: 1998-091662/ 199809

XRPX Acc No: N98-072895

Programmable controller for industrial apparatus - includes device memory  
whose contents is divided into various device range according to index  
value set by index register

Patent Assignee: MITSUBISHI ELECTRIC CORP (MITQ )

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
JP 9319410	A	19971212	JP 96140189	A	19960603	199809 B

Priority Applications (No Type Date): JP 96140189 A 19960603

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 9319410	A		8	G05B-019/05	

Abstract (Basic): JP 9319410 A

The controller (1) has a sequence program memory (6) to store the  
program in sequence. A CPU (2) has index register (11) to set an index  
value to each device.

The contents stored in a device memory (3) is divided into various  
device range according to the index value.

ADVANTAGE - Increases speed of device process. Uses device of large  
capacity.





## 【特許請求の範囲】

【請求項1】 記憶されたシーケンスプログラムに基づき、産業機器等の制御対象機器を制御するプログラマブルコントローラにおいて、前記シーケンスプログラムのデバイスを命令コード及びインデックス領域としてインデックスを付加されたデバイスコードに変換した形で格納するシーケンスプログラムメモリと、所定のアドレス毎に区切られて使用されるように構成されたデバイスメモリと、前記プログラム毎のローカルデバイスのアドレスの値がセットされるインデックスレジスタを有するCPUと、前記デバイスコードから前記インデックスレジスタにセットされているローカルデバイスのアドレスの値を参照して前記デバイスメモリをデバイス範囲に区切った内容をローカルデバイスの内容とし、その内容によりデバイスを処理することを特徴とするプログラマブルコントローラ。

【請求項2】 デバイスコードにプログラムNO.を指定する箇所を付加したことにより、そのプログラムで使用できるローカルデバイスを決定することを特徴とする請求項1に記載のプログラマブルコントローラ。

【請求項3】 デバイスをブロックにわけてアクセスするためにデバイスコードにおいてデバイスのブロックサイズを決める命令表現と命令コードを作成することにより、プログラム中で前記ブロックサイズを決定することを特徴とする請求項1又は請求項2に記載のプログラマブルコントローラ。

【請求項4】 デバイスコードにブロックNO.を指定する箇所を付加したことにより、指定したブロックNO.でデバイスを使用することを特徴とする請求項3に記載のプログラマブルコントローラ。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】この発明は、産業機器などで使用されるプログラマブルコントローラに関するもので、特にシーケンスプログラムのデバイス処理のインデックス修飾に関するものである。

## 【0002】

【従来の技術】従来のこの種のプログラマブルコントローラにおけるデバイス処理（ローカルデバイス処理）を説明するものとして図11～図15を用いて説明する。

【0003】図11は、従来のプログラマブルコントローラの全体の構成図である。1は、プログラマブルコントローラで、2は、このプログラマブルコントローラ1のCPUである。3は、プログラマブルコントローラ1で処理されるプログラムでD0、D1といったデバイスデータを扱うためのデバイスメモリである。4は、デバイスメモリ3と同様にCPU2からアクセスできる増設メモリである。5は、プログラマブルコントローラ1に接続し、ユーザがシーケンスプログラムを作成したりプログラマブルコントローラ1にシーケンスプログラム

を格納するための周辺装置である。6は、ユーザが作成したシーケンスプログラムを命令コード、デバイスコードに変換した形で格納するためのシーケンスプログラムメモリである。

【0004】シーケンスプログラムメモリ6は、プログラム1用、プログラム2用というように分割されたエリアになっている。増設メモリ4は、プログラム1で使用しているデバイス用エリア7、プログラム2で使用しているデバイス用エリア8というように分割されている。

【0005】図12は、図11のプログラマブルコントローラ1におけるCPU2のシーケンスプログラムを処理するときのデバイス処理に関する部分の処理フローである。

【0006】図13は、シーケンスプログラムのデバイスD0を扱う命令の例をラダー図で示したものであり、デバイスD0の値をこの命令131が実行するたびに+1してD0に格納するシーケンスプログラムである。

【0007】図14は、図13でのシーケンスプログラムでの命令131でのデバイスD0のデバイスコードの構成を示したものである。この構成で図11のシーケンスプログラムメモリ6に格納されている。141はデバイス判定、142はアドレスを示すためのエリアである。

【0008】図15は、プログラマブルコントローラを使用した工場のラインを示す説明図である。1は、ラインのプログラマブルコントローラである、9は第1ライン、10は第2ラインである。

【0009】ここで、従来の技術の具体的な例を手順にそって説明する。図15にあるような工場の第1ライン9、第2ライン10を考える。例えば、2つの生産ラインをプログラマブルコントローラ1台で監視して、動作状況における計量を管理しようとする。この場合、ユーザはプログラマブルコントローラ1に第1ライン9用のプログラムAと第2ライン10用のプログラムBを用意する。この2つのプログラムは、それぞれのラインにおける物体が流れた数を累積して、その数を管理しようとするものである。そして、その累積の数をデバイスD0に入れていくことにする。この2つのプログラムは物体の数を累積していくということでは処理が同じため、プログラムAを作成し、それをコピーする形でプログラムBを作成できる。よって、どちらのプログラムも累積の数をデバイスD0に入れていくことになる。

【0010】通常はプログラムによって、プログラムAはデバイスD0、プログラムBはデバイスD1というように、プログラムによってデバイスを変えるのだが上記のようにプログラムAとプログラムBで計量を累積するという点で処理が同様であるため、ユーザはプログラムAをコピーしてプログラムBを作成することができる。そこで、第1ライン9でさらにもう1つデバイスを使用するようにプログラムを変更してデバイスD2を使用す

るか、デバイスD1を使用しプログラムBのデバイスD1をデバイスD2に変更するという作業が発生する。従って、プログラム間で、どのデバイスを使用するかという取り決めが必要でありプログラムの作成変更が複雑になってくる。また、現在では使用するデバイスが多くなってきており、例えば一つのプログラムで100近くのデバイスを使い、そのようなプログラムが100近くあるとD54321というようにデバイスのアドレスが大きくなってしまい、ユーザがプログラム中で扱いにくくなってきている。そのため、同じデバイスをそのまま使用できるようにするための処理がローカルデバイス処理であり、その処理を以下に説明する。

【0011】前例でユーザがシーケンスプログラムA、シーケンスプログラムBの2本のプログラムを作成したものとする。シーケンスプログラムA、シーケンスプログラムBにはそれぞれ同様に図13に示す命令が作成されているものとする。この命令はINC命令で、これはデバイスD0の値を+1してD0に格納するシーケンスプログラムである。図11のシーケンスプログラムメモリ6にあるプログラム1が実行される場合、まずローカルデバイスを復帰させる処理として増設メモリ4にあるプログラム1用のデバイスエリア7の内容をデバイスメモリ3にOSが転送する。図12のフローではステップ122にあたる。そして、プログラム1が実行され、そのなかでD0の値を処理する場合、デバイスメモリ3のAにあるD0の内容を図14のデバイスコードから参照して処理する。

【0012】命令処理時のデバイスD0の処理では、図13のシーケンスプログラムのデバイスD0をデバイスコードで示すと図14のようになる。図14の141はデバイス判別を表し、142はアドレス部である。このデバイスコードは図11の周辺装置5によってラダー図から変換し、シーケンスプログラムメモリ6に格納される。そして、このコードからH/Wがデバイスメモリ3のAをこのデバイスが使用することを認識する。

【0013】次に、ローカルデバイスを退避させるためにデバイスメモリ3のデバイスD0の内容Aを増設メモリ4のプログラム1用の箇所にOSが転送する。これが図12のフローではステップ124にあたる。図12のフローのステップ125で次に実行するプログラムがあるかの判断をしている。ある場合は次のプログラムのデバイスの復帰をOSがおこない、無い場合は1通りのプログラム実行は終了し他の処理を実行する。次に実行するプログラムがある場合のデバイスのOSによる復帰、退避はプログラム1の場合と同様である。

【0014】以上のくり返しによって複数プログラムで同じデバイス名、すなわち同じデバイスコードをもちいても別のデバイスとして扱うことができる。このOSが行う復帰、参照、退避によって上記の例のような容易に複数のプログラムを作成するときにおきる同一デバイス

の使用を気にせず作成が可能となる。

【0015】

【発明が解決しようとする課題】上述のように、従来のプログラマブルコントローラにおけるシーケンスプログラムでのローカルデバイス処理は、複数シーケンスプログラムの各プログラム実行前に増設メモリ4にあるデバイスデータをデバイスメモリ3に復帰させ、プログラム実行後にデバイスメモリ3のデバイスデータを増設メモリ4に退避させるOS側での処理が必要であり、そのため復帰、退避の処理の時間が必要であるため、シーケンスプログラムの処理時間がのび、本来の入出力制御のレスポンスが遅くなる。

【0016】現在では、プログラム中で使用するデバイスが多くなっており、また大容量のデバイスを用いるとアドレスが大きくなりデバイスD34567のようにユーザがプログラムを作成するにあたり、扱いにくさが発生している。

【0017】この発明では、従来のローカルデバイス処理で必要になっていた時間を短縮し、ローカルデバイス処理を高速に行うこと、デバイスコードに使用するプログラムNO.、使用するデバイスのブロックNO.を指定することにより、大容量のデバイスを分別して使用する際に容易にデバイス処理が行うことができるプログラマブルコントローラを得ることを目的としたものである。

【0018】

【課題を解決するための手段】この発明に関するプログラマブルコントローラは、記憶されたシーケンスプログラムに基づき、産業機器等の制御対象機器を制御するプログラマブルコントローラにおいて、前記シーケンスプログラムのデバイスを命令コード及びインデックス領域としてインデックスを付加されたデバイスコードに変換した形で格納するシーケンスプログラムメモリと、所定のアドレス毎に区切られて使用されるように構成されたデバイスメモリと、前記プログラム毎のローカルデバイスのアドレスの値がセットされるインデックスレジスタを有するCPUと、前記デバイスコードから前記インデックスレジスタにセットされているローカルデバイスのアドレスの値を参照して前記デバイスメモリをデバイス範囲に区切った内容をローカルデバイスの内容とし、その内容によりデバイスを処理するものである。

【0019】また、デバイスコードにプログラムNO.を指定する箇所を付加したことにより、そのプログラムで利用できるローカルデバイスを決定するものである。

【0020】また、デバイスをブロックにわけてアクセスするためにデバイスコードにおいてデバイスのブロックサイズを決める命令表現と命令コードを作成することにより、プログラム中で前記ブロックサイズを決定するものである。

【0021】また、デバイスコードにブロックNO.を

指定する箇所を付加したことにより、指定したブロックNO.でデバイスを使用するものである。

#### 【0022】

【発明の実施の形態】この発明の実施の形態を図1から図10を用いて説明する。図1は、この発明におけるプログラマブルコントローラの構成図である。1は、プログラマブルコントローラ、2は、プログラマブルコントローラ1のCPUである。3は従来例と同様のデバイスメモリである。5は、プログラマブルコントローラ1に接続し、ユーザがシーケンスプログラムを作成したりプログラマブルコントローラ1にシーケンスプログラムを格納するための周辺装置である。6は、ユーザが作成したシーケンスプログラムを命令コード、デバイスコードに変換した形で格納するためのシーケンスプログラムメモリである。シーケンスプログラムメモリ6には、プログラム1とプログラム2がある。11は、CPU2に設けられ、この発明により追加するインデックスレジスタである。

【0023】実施の形態1. 図2は、この発明の実施の形態1によるプログラマブルコントローラのデバイス処理フローである。従来例と違うのはステップ122、124がステップ22になっていることである。

【0024】図3は、この発明の実施の形態1によるプログラマブルコントローラのシーケンスプログラムで、デバイスD0を扱う命令の例をラダー図31で示したものであり、命令としては図13と同様のINC命令である。

【0025】図4は、図3でのシーケンスプログラムでのデバイスD0のデバイスコードの構成を示したものである。

【0026】図5は、図1のデバイスメモリ3をOSによって区切られたデバイスメモリの状態である。

【0027】実施の形態1として、具体例を挙げて説明する。図3のようなシーケンスプログラムを図1の周辺装置5によってユーザが作成する。そして、周辺装置5がそのシーケンスプログラムのデバイスD0をデバイスコードに変換し、図1のプログラマブルコントローラ1のシーケンスプログラムメモリ6に格納する。格納されたデバイスコードの構成が図4である。41は、図14の141と同様にデバイス判定を示している。43も142と同様にデバイスのアドレスを示している。42がこの発明で追加したインデックスIDXである。

【0028】この図4のデバイスコードのインデックス42には図1のプログラマブルコントローラ1のCPU2に、この発明によって付加したインデックスレジスタ42を参照する箇所を表す値を周辺装置5がセットする。このインデックスの値はあらかじめ周辺装置5とCPU2との間の取り決めで、ローカルデバイスのインデックスレジスタ11を参照するかを決めておく。例えば図1のインデックスレジスタ11のZ5を使用する

と決める。

【0029】具体例をフローで示すと図2のようになる。まず、ステップ21でユーザが作成したプログラムのNO.を変数mにセットする。次にステップ22でローカルデバイスの実アドレスをOSがセットする処理を次のようにして行う。

【0030】この処理は、図1のデバイスメモリ3においてプログラム1が使用するデバイス範囲が0から100だとする。この範囲からOSがプログラム1のデバイスD0が指すアドレスを計算する。デバイスメモリ3は、図5に示すように100番地毎に区切られて使用されるように構成されており、プログラム1でのデバイスD0のアドレスはOSによって0Hとなる。そして、この0Hの値をOSが図1のインデックスレジスタ11のZ5の箇所にセットする。

【0031】次に図2のフローのステップ23でプログラムを実行する。図3のシーケンスプログラムで実際にデバイスD0を処理するときは、図4のデバイスコードから図1の周辺装置5がセットしたIDX42のZ5を示す値によって図1のインデックスレジスタ11のZ5の箇所をOSが参照してZ5の箇所にセットされている0HがデバイスD0のアドレスと決まる。そしてH/Wが図1のデバイスメモリ3をデバイス範囲に区切った図5の51の内容をデバイスD0の内容とするのである。

【0032】ステップ24で次のプログラムがあるかどうかの判定をしている。あれば、次のプログラムで使用しているデバイスについて上記と同様のことを行い、プログラム中でデバイスの処理をおこなう。図5のデバイスメモリの例においてプログラム2のデバイスD0は100H、プログラム3のデバイスD0は200Hとなる。

【0033】この実施の形態1では、デバイスコードにインデックス領域としてIDXを付加する。従って周辺装置でユーザが作成したシーケンスプログラムを新規デバイスコードに変換する。CPUは新規のデバイスコードを処理するとき、ローカルデバイス用インデックスを行うようにする。CPUがプログラム実行前にインデックスにデータをセットする。

【0034】以上により、デバイスコードにローカルデバイス用インデックス修飾を設けるため、従来のようにOSによって行われていたデバイスデータの増設メモリへの退避、増設メモリからの復帰をする必要がなくなりデバイス処理が高速になる。

【0035】実施の形態2. 図6は、シーケンスプログラムのデバイスD0を扱う命令の例をラダー図で示したものである。INC命令については図13と同様である。

【0036】図7は、図6でのシーケンスプログラムの命令61でのデバイスD0のデバイスコードの構成を表したものである。

【0037】次に具体的な例を用いて説明する。まず、図6にあるシーケンスプログラムを図1の周辺装置5によって作成する。次に周辺装置5がシーケンスプログラムをデバイスコードに変換し図1のプログラムコントローラ1に格納する。その格納されたデバイスコードの構成を図7に示す。図7のデバイスコードは、図4のデバイスコードに71を加えたものである。コード71の73は図4のデバイスコード同様にデバイス判定である。ここがPになっていることで以降のプログラムNO.を指定するコードということになる。72には実際のプログラムNO.がセットされる。これは、図6のようなシーケンスプログラムを図1の周辺装置5がデバイスコードに変換する際に周辺装置5がセットするのである。この74に周辺装置5がセットした値によってOSがプログラムNO.2で使用するデバイスD0ということを確認する。図7のデバイスコードの75のコードによって実施の形態1と同様に図1のデバイスメモリ3を図5のようにプログラムごとのD0の先頭アドレスで区切る。この実施の形態2の場合もデバイス範囲は100Hとする。

【0038】75のコードによって実施の形態1と同様にデバイスD0のアドレスをきめるのだが、実施の形態2では71のコードがついているため、図1のデバイスメモリ3を区切った図5のデバイスメモリでプログラム2のデバイスD0のアドレス、すなわち100HがデバイスD0のアドレスとOSによってきめられる。この後の図1のシーケンスプログラムメモリ6にある図7のデバイスコードがインデックスレジスタ11を参照しプログラム中でデバイスメモリの内容を処理する方法は実施の形態1の例と同様にOS、H/Wが行う。

【0039】なお、図7のデバイスの75では、実施の形態1で示した図1の周辺装置5とCPU2とでの取り決めは、インデックスレジスタ11のZ4を参照することによってきめておく。

【0040】以上により、複数のプログラムで同デバイスを使用する場合でも、従来のようにデバイスデータの増設メモリへの退避、増設メモリからの復帰をする必要がなくなりデバイス処理が高速になる。

【0041】実施の形態1、実施の形態2でデバイスメモリ3での1プログラムに使用できるデバイス範囲を決めていたが、デバイスメモリ3をいくつかのブロックにわけてブロックごとにアクセスをするようにしたのが実施の形態3、実施の形態4である。

【0042】実施の形態3。図8は、ユーザが図1の周辺装置5で作成するシーケンスプログラムである。ラダー図81がブロックサイズ指定の命令、ラダー図82、ラダー図83はINC命令である。INC命令については図13と同様である。

【0043】図9は、図1のデバイスメモリ3をブロックサイズによって区切ったデバイスメモリの状態であ

る。

【0044】図8のラダー図81によって図1のデバイスメモリでの1ブロックのサイズを決めることができる。BSIZE84は図1のデバイスメモリ3をブロックにわける命令であり、K5の箇所85で1つのブロックサイズを指定する。ラダー図81ではデバイスメモリ3を図9のように5Hごとに区切ったことになる。図8のラダー図82、83は図3と同様のINC命令のラダー図である。このラダー図82の86にあるようにB2と指定することでブロックNO2を表し、ラダー図82はブロックNO2のデバイスR0の値を+1してブロックNO2のデバイスR0にいれることになる。ラダー図83も同様であるが、87でB1となっていることからブロックNO1のデバイスR0の値を+1してブロックNO1のデバイスR0にいれることになる。

【0045】以上により、ユーザが大容量のデバイスを分別して使用するプログラム中でブロックサイズを決めることができる。

【0046】この図8のラダー図83のデバイスコードのアクセス法が実施の形態4である。

【0047】実施の形態4。図10は、図8のラダー図82を図1の周辺装置5が変換し、シーケンスプログラムメモリ6に格納したデバイスコードである。

【0048】次に具体的な例を用いて説明する。まず、図8のようなシーケンスプログラムをユーザが図1の周辺装置5で作成し、周辺装置5が図1のプログラマブルコントローラ1のシーケンスプログラムメモリ6に格納する。そのデバイスコードが図10のデバイスコード101である。このデバイスコード101は先の実施の形態1でのデバイスコード103にコード102を付加したものである。

【0049】コード102はブロックサイズを指定するコードであり、デバイス判別がBになっていることからブロックサイズの指定ということを確認する。そして、105にある2によってブロックNO2を指定していることになる。106のB、105の2は図1の周辺装置5がセットする。このコード102によってOSがブロックNO2ということを確認する。

【0050】図8のユーザが作成したラダー図81ではブロックサイズが5となっているため図1のデバイスメモリ3は図9のようになる。そして、図10のデバイスコード101のコード103から図1のインデックスレジスタ11の値をOSが参照しデバイスコード101のアドレスがきまる。ここでは、図1のデバイスメモリ3が図9のようになることにより92の部分が使用できるデバイス範囲となる。

【0051】図1のインデックスレジスタ11に値をセットすることは、実施の形態1と同様であり、コード103の104に3となっているのは、ここに図1のインデックスレジスタ11のZ3を示す値がセットされてい

る。これは実施の形態1と同様にOSがおこない、図1の周辺装置5とCPU2との取り決めでZ3を使用するとなっているからである。この後のプログラム中でのデバイスR0は実施の形態1、実施の形態2と同様に図1のデバイスメモリ3の内容を処理する。

【0052】以上により、指定したブロックNOでのデバイスを使用することになり大容量のデバイスを用いるとアドレスが大きくなりユーザにとって扱いやすくなる。

【0053】

【発明の効果】以上のように、この発明によれば、記憶されたシーケンスプログラムに基づき、産業機器等の制御対象機器を制御するプログラマブルコントローラにおいて、シーケンスプログラムのデバイスを命令コード及びインデックス領域としてインデックスを付加されたデバイスコードに変換した形で格納するシーケンスプログラムメモリと、所定のアドレス毎に区切られて使用されるように構成されたデバイスメモリと、プログラム毎のローカルデバイスのアドレスの値がセットされるインデックスレジスタを有するCPUと、デバイスコードからインデックスレジスタにセットされているローカルデバイスのアドレスの値を参照して前記デバイスメモリをデバイス範囲に区切った内容をローカルデバイスの内容とし、その内容によりデバイスを処理することにより、デバイスコードにローカルデバイス用インデックス修飾を設けるため、従来のようにOSによって行われていたデバイスデータの増設メモリへの退避、増設メモリからの復帰をする必要がなくなりデバイス処理が高速になる効果がある。

【0054】また、デバイスコードにプログラムNO.を指定する箇所を付加したことにより、そのプログラムで使用できるローカルデバイスを決定することにより、複数のプログラムで同デバイスを使用する場合でも、従来のようにデバイスデータの増設メモリへの退避、増設メモリからの復帰をする必要がなくなりデバイス処理が高速になる効果がある。

【0055】また、デバイスをブロックにわけてアクセスするためにデバイスコードにおいてデバイスのブロックサイズを決める命令表現と命令コードを作成することにより、ユーザが大容量のデバイスを分別して使用するプログラム中でブロックサイズを決めることができる。

【0056】また、デバイスコードにブロックNO.を指定する箇所を付加したことにより、指定したブロックNO.でのデバイスを使用することになり大容量のデバイスを用いるとアドレスが大きくなりユーザにとって扱いやすくなる効果が得られる。

【図面の簡単な説明】

【図1】 この発明によるプログラマブルコントローラの構成図である。

【図2】 この発明の実施の形態1でのプログラム実行でのデバイス処理を示すフローチャートである。

【図3】 この発明の実施の形態1でのシーケンスプログラムを示す図である。

【図4】 図3のシーケンスプログラムでのデバイスD0のデバイスコードを示す説明図である。

【図5】 この発明の実施の形態1によるプログラマブルコントローラのデバイスメモリを示す構成図である。

【図6】 この発明の実施の形態2でのシーケンスプログラムを示す図である。

【図7】 図6のシーケンスプログラムでのデバイスD0のデバイスコードを示す説明図である。

【図8】 この発明の実施の形態3、4でのシーケンスプログラムを示す図である。

【図9】 この発明の実施の形態3、4によるプログラマブルコントローラのデバイスメモリを示す構成図である。

【図10】 図8のシーケンスプログラムでのデバイスR0のデバイスコードを示す説明図である。

【図11】 従来のプログラマブルコントローラの構成図である。

【図12】 従来のプログラム実行でのデバイス処理を示すフローチャートである。

【図13】 従来のプログラマブルコントローラでのシーケンスプログラムを示す図である。

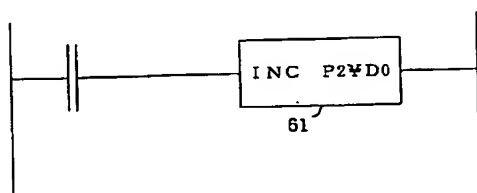
【図14】 図13のシーケンスプログラムでのデバイスD0のデバイスコードを示す説明図である。

【図15】 プログラマブルコントローラを使用したラインを示す構成図である。

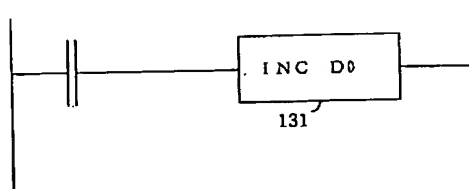
【符号の説明】

1 プログラマブルコントローラ、2 CPU、3 デバイスメモリ、5 周辺装置、6 シーケンスプログラムメモリ、11 インデックスレジスタ。

【図6】

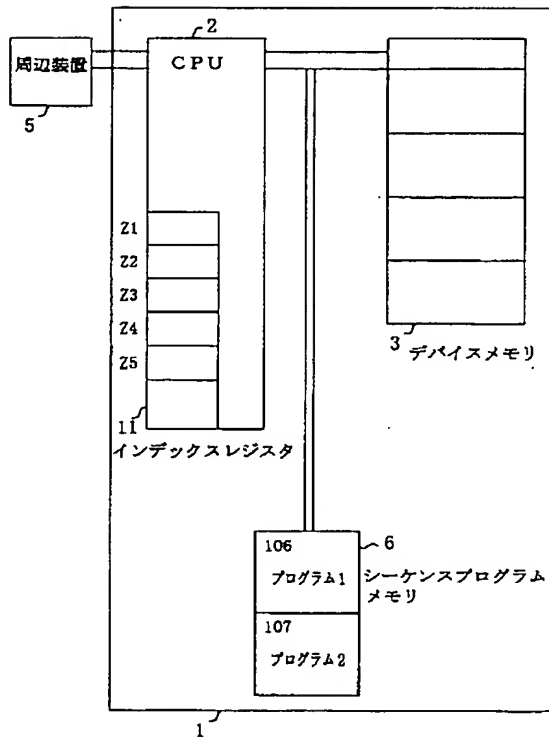


【図13】

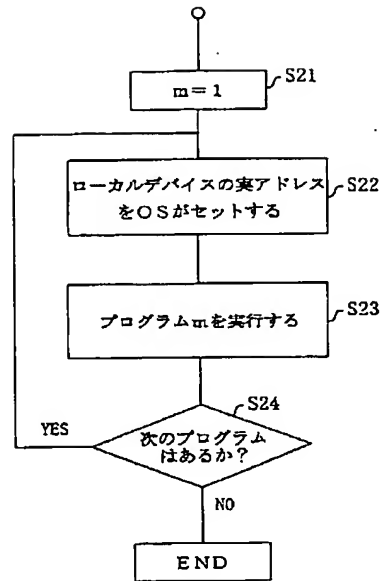




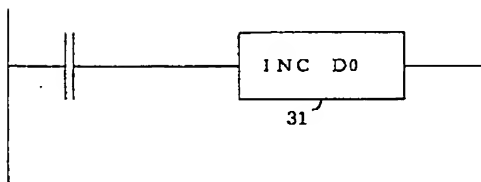
【図1】



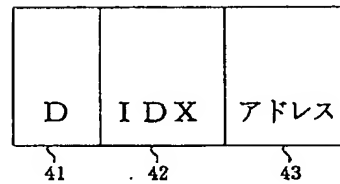
【図2】



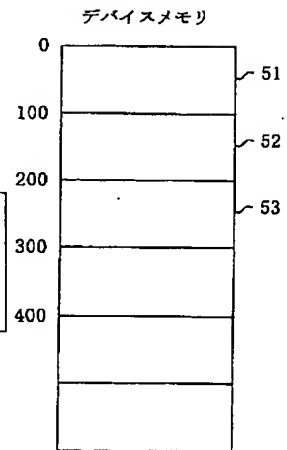
【図3】



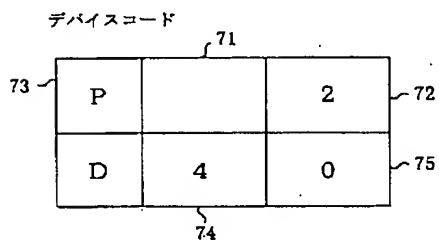
【図4】



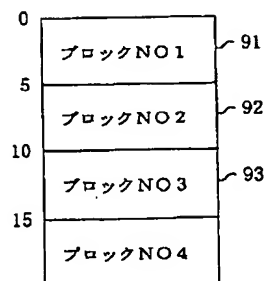
【図5】



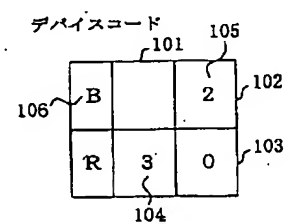
【図7】



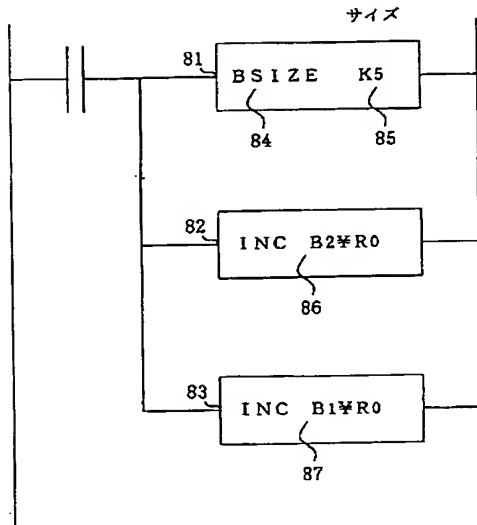
【図9】



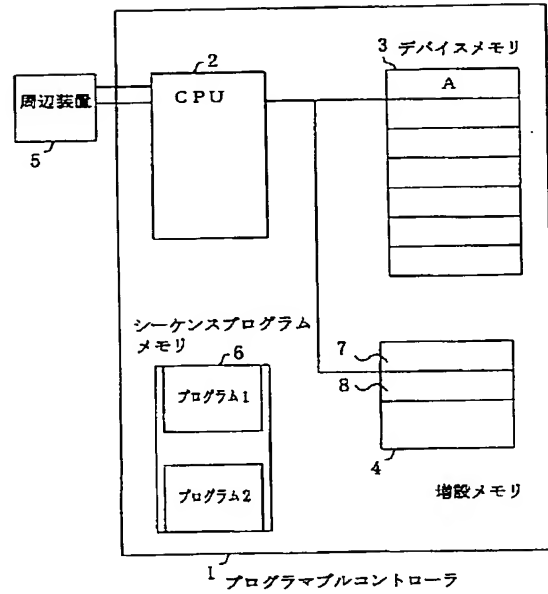
【図10】



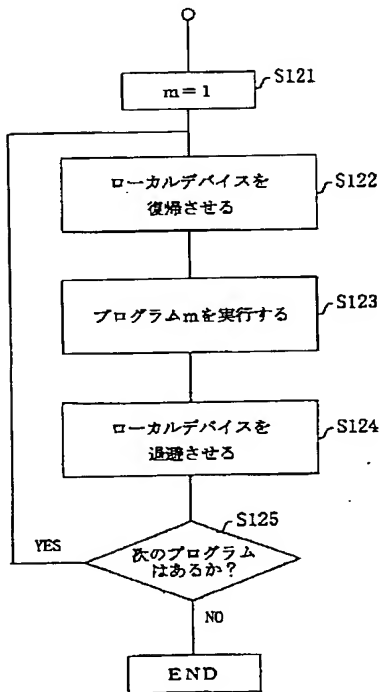
【図8】



【図11】

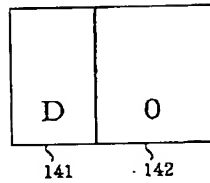


【図12】

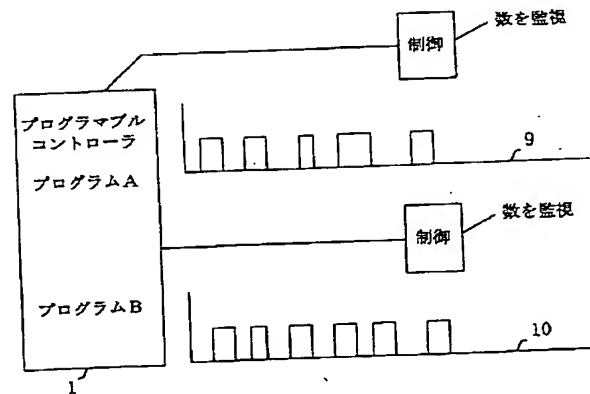


【図14】

デバイスコード



【図15】



□ はラインを流れる物体